Remote Access to Very Large Image Repositories
A High Performance Computing Perspective

Lucian Plesea

Jet Propulsion Laboratory

High Performance Image Access

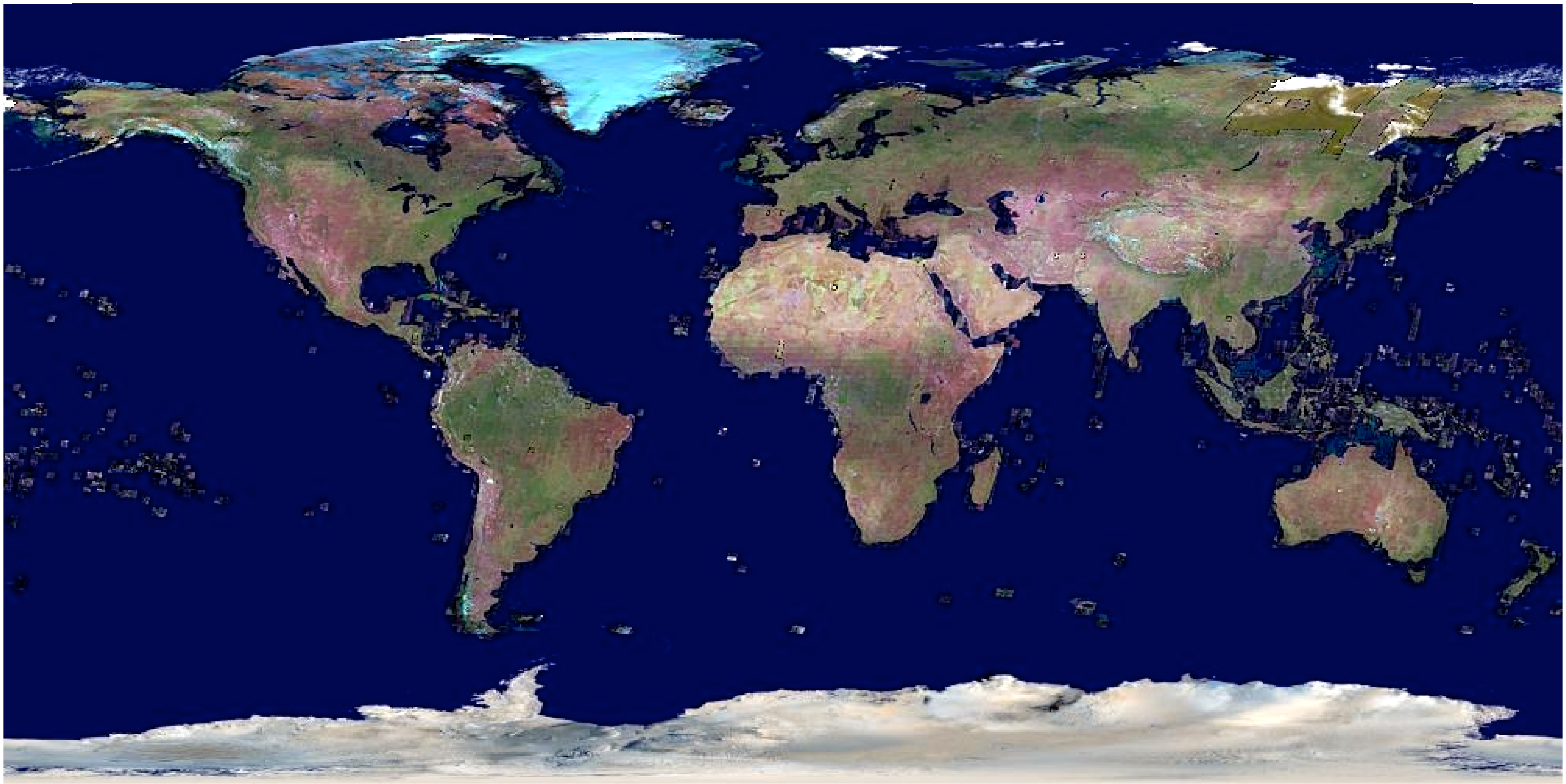
- When dataset sizes are very large, the performance of data access becomes essential
- But the problem is not just data access, every aspect of dealing with data has to be addressed

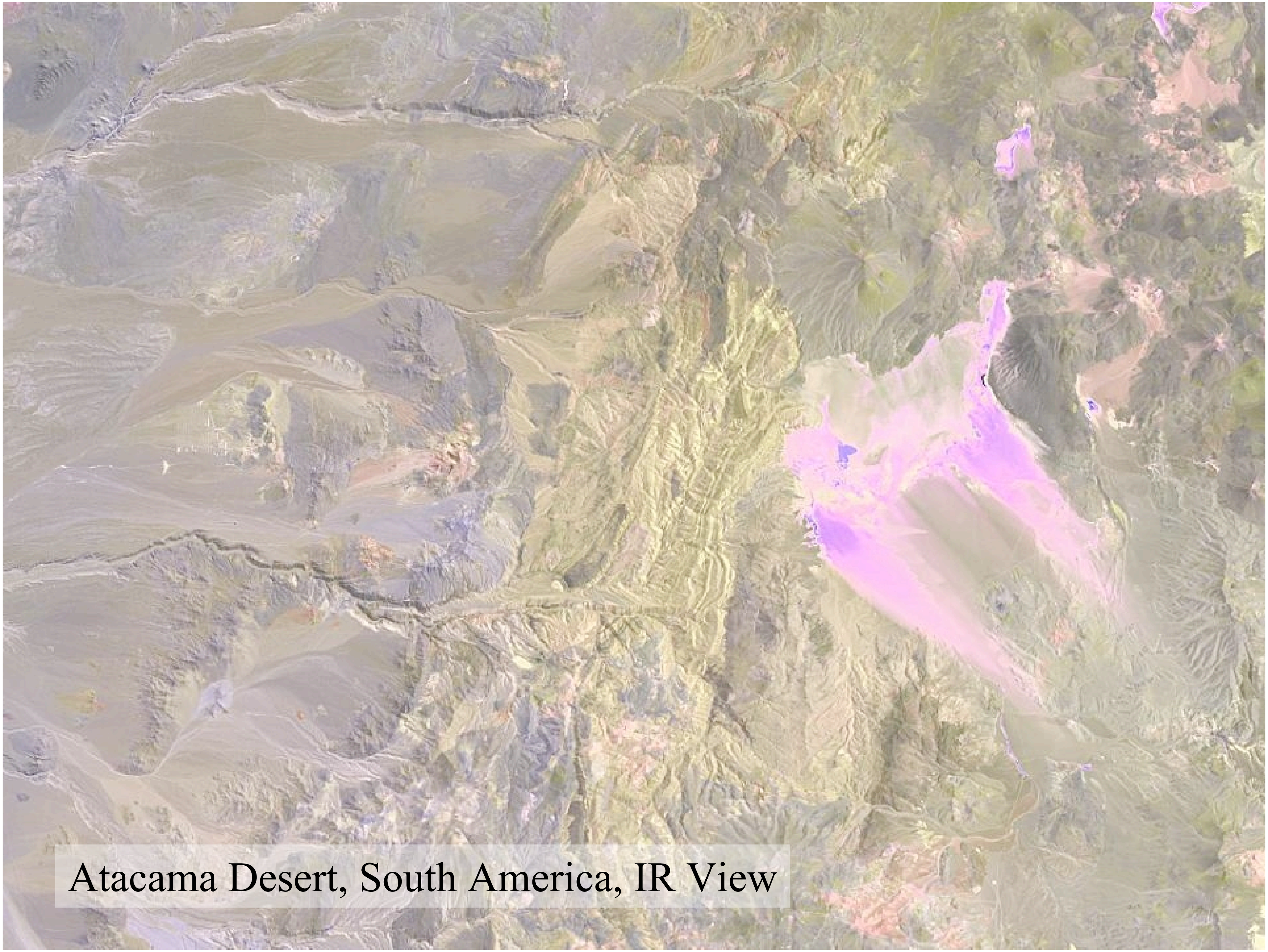
  Storage; storage file format; data access; remote data access; access to output data while processing
- The main example used is the project to produce the WMS Global Mosaic and an associated WMS server.

WMS Global Mosaic

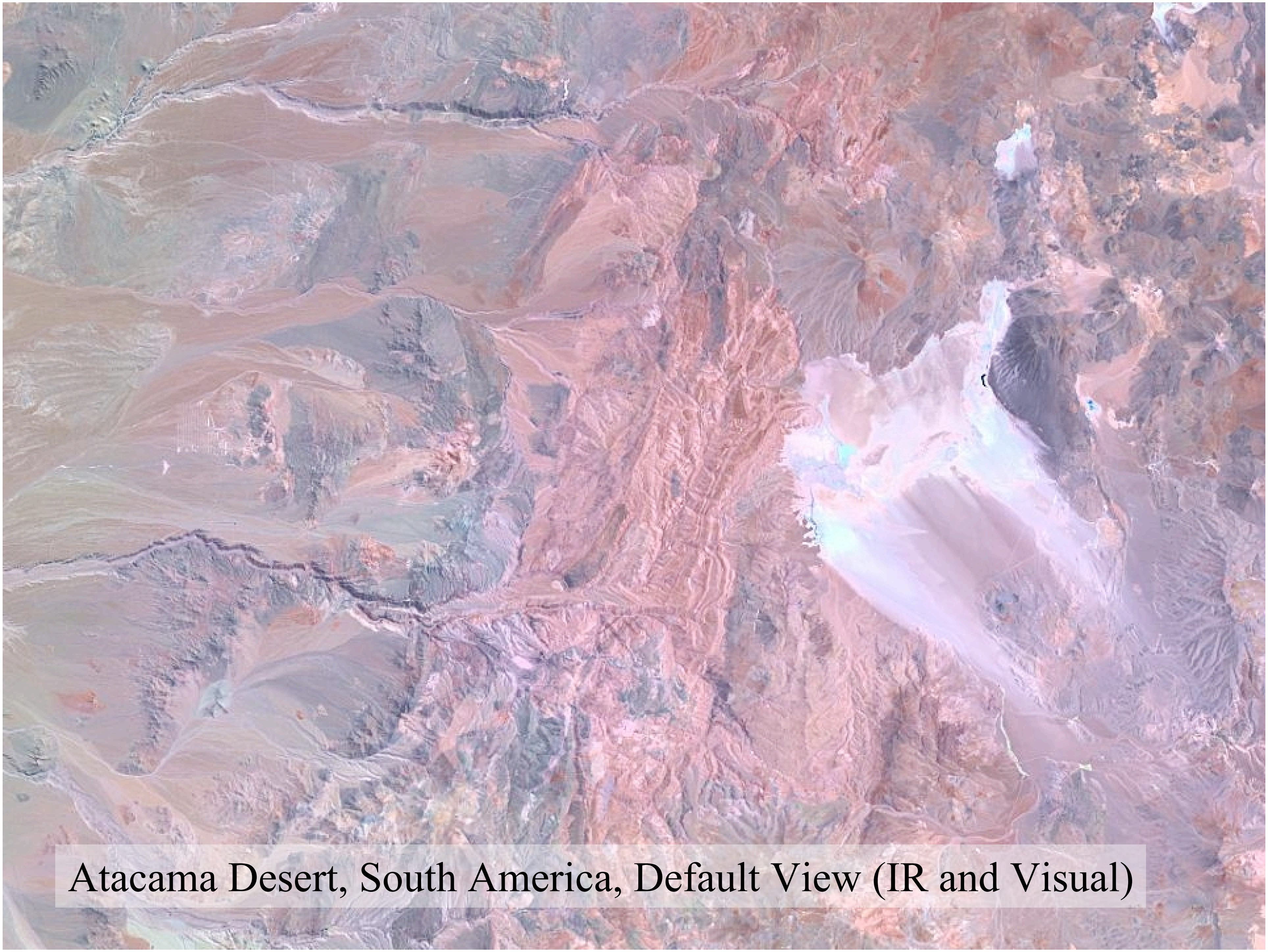
- A composite image built from 8208 full resolution Landsat 7 scenes
- This mosaic has 3600 times more pixels than previous global earth images, at 15m per pixel
- This mosaic contains about three trillion pixels, and more than five Terabytes of data.
- OnEarth is the name of the JPL Web Map Service (WMS) server, hosting a number of high resolution earth image datasets

WMS Global Mosaic

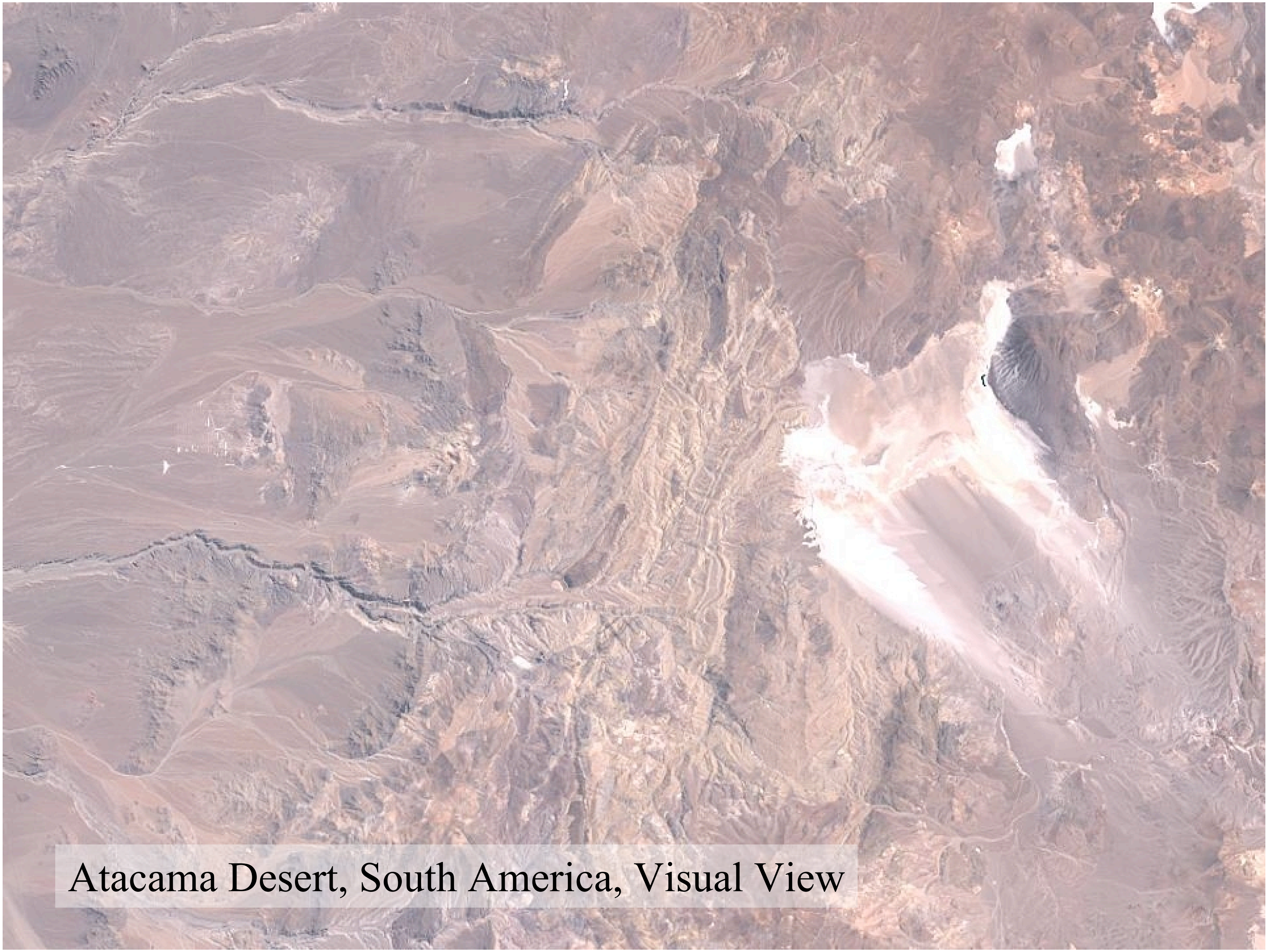




Atacama Desert, South America, IR View



Atacama Desert, South America, Default View (IR and Visual)



Atacama Desert, South America, Visual View

The Mosaic Project

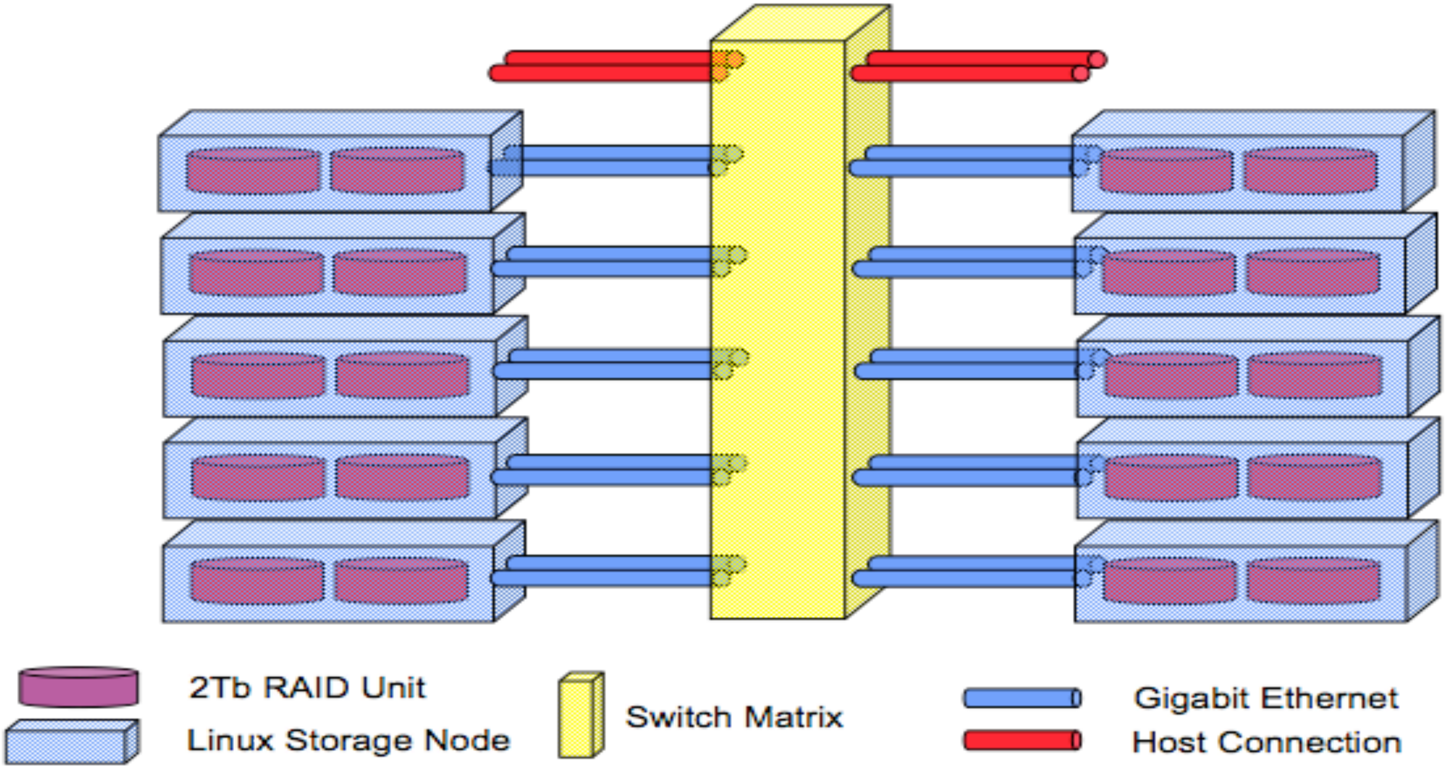
- NASA GIO is leading an effort to increase the interoperability of GIS tools and remote imagery availability.
- The availability of a recent, high resolution, global coverage map of the earth was seen as an important component of this effort.
- Release One, April 2004
- Beta test of final release, on-line June 2005
- Will be soon available for distribution as GeoTiff files

Storage

- For this project and a few others, a very large storage system was necessary.
- Raid Again Storage using Commodity Hardware And Linux RASCHAL, a 40TB NAS storage cluster, built in-house.
- RASCHAL became operational April 2003, and has been in continuous use since then.
- OnEarth uses about 20 TBytes

Storage

RASCHAL System Diagram



File Format

- Most existing image file formats have significant size and access limitations, most are 32 bit formats and cannot efficiently handle access to local areas.
- A Journaling File format was used, a tiled, multi-spectral and multi-resolution file format that supports lossless and lossy compression at the tile level. It is intended for use as a high performance, “work in progress” storage format and not for distribution.
- A level of indirection in data access, adding journaling features which ensures file consistency.
- Uses external metadata, which increases performance and flexibility at the expense of user friendliness.

Remote Data Access

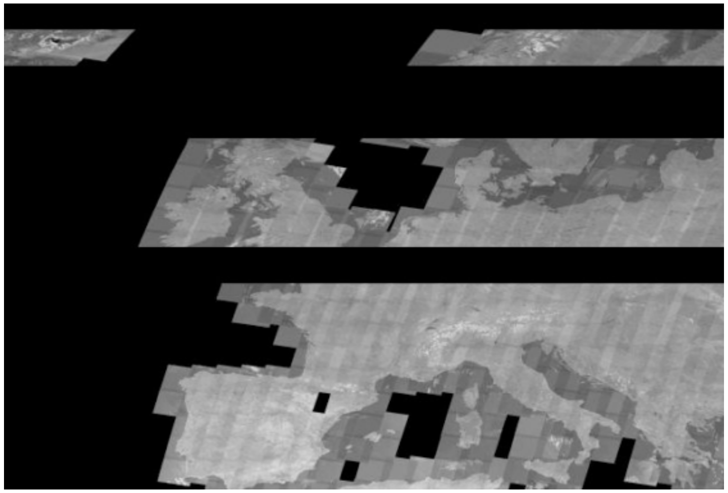
- To create the WMS Global Mosaic, the Network Image Protocol was used, separating the location and specific file format from the application
- Computation resources are located remote from the storage resources
- Data staging is prohibitively expensive or might not even be an option
- Targeted computation system might change during the processing
- An image specific data access subsystem allows regions of the input and output images to be transferred independently

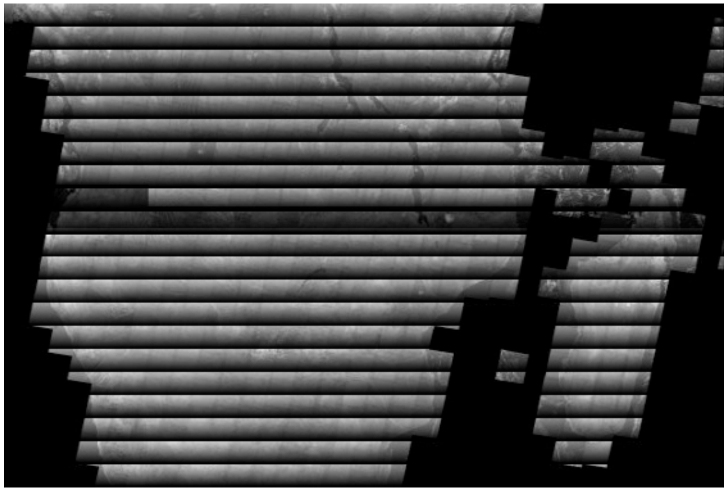
Remote Data Access

- Processing of the GeoCover 2000 into the WMS Global Mosaic took about 21 days
- During this time, the sustained data rate was 2.5MB/sec read and 1.5MB/sec write. Peak rates were close to 15MB/sec, the available bandwidth
- Three separate Linux image servers were used for data input, and two more were used as output repositories
- Performance scaled linearly with the number of computation CPUs used, even when compared with the local execution model
- Secure Socket Layer implementation incurs significant overhead.

Real Time Access

- For very large datasets, it is essential to be able to inspect the results while the processing is still being done
- In the WMS case, the matrix of the size field of the compressed tile can be visualized as an image
- The size after compression encapsulates the entropy information about each specific region

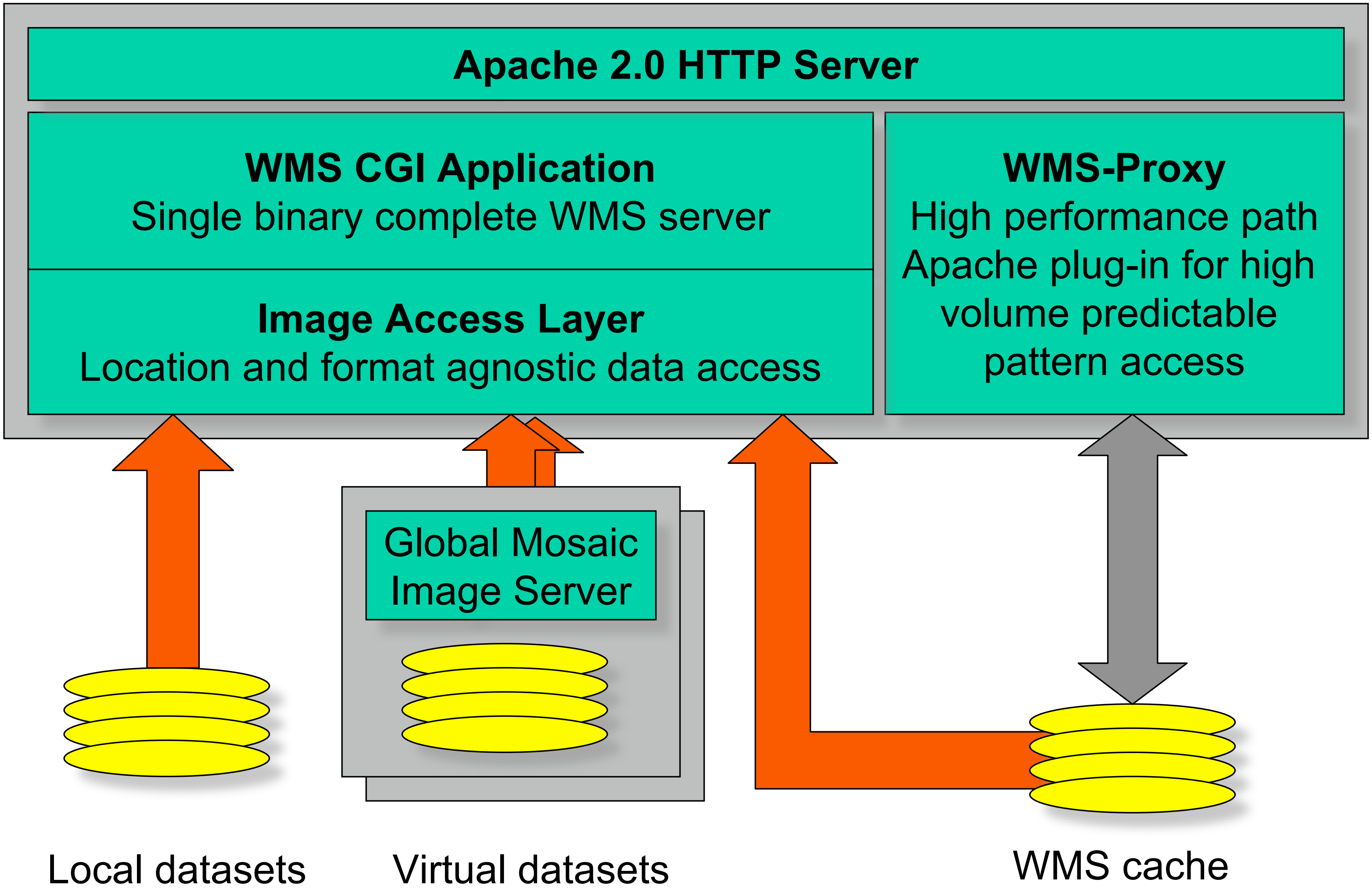




Access: WMS

- A high performance WMS server can be used as a data source.
- The OnEarth WMS server is implemented as a CGI application, and uses the same technologies as the mosaic application.
- Server provides many basic operatons, such as color selection, pan-sharpening, multiple projections.
- Image control using Styled Layer Descriptor.

OnEarth WMS Server Architecture



Apache 2.0 HTTP Server

WMS CGI Application
Single binary complete WMS server

Image Access Layer
Location and format agnostic data access

WMS-Proxy
High performance path
Apache plug-in for high volume predictable pattern access

Global Mosaic Image Server

Local datasets

Virtual datasets

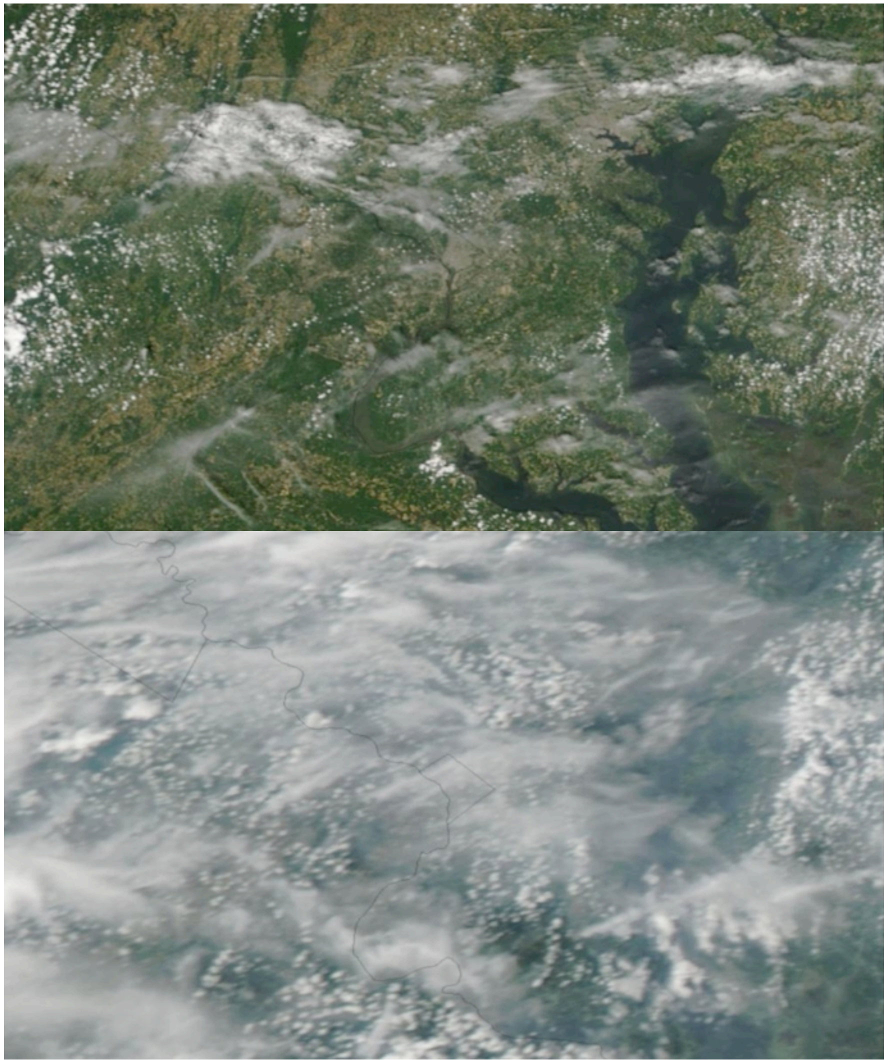
WMS cache

Access: WMS

- High performance use is available via a WMS proxy, which uses the same WSM protocol but caches a certain class of data.
- Small latencies and good support due to use of HTTP.
- On the current deployed server, the main limitation is the disk seek time, with more than 150 data transactions per second being measured.
- Has the benefit of providing customized data.

Another Example: Daily MODIS

- Two daily composites, with global coverage at 250m resolution are being built every night, from image data produced by the Rapid Response MODIS team
- These composites are built directly into a jpeg compressed format used by the WMS proxy



WMS Client Applications

- **WorldKit.com, Mapufacture.com**
  - Flash based clients that can be embedded in a web page.
- **Intergraph WMS viewer, http://www.wmsviewer.com**
  - Windows based freeware browser
- **EDINA Map Viewer, http://geoview.edina.ac.uk/JPL/index.jsp**
  - Public access, educational
- **Earth Sun Gateway, http://esg.gsfc.nasa.gov**
  - NASA GIS interface
- **Mapserver, http://mapserver.gis.umn.edu/**
  - Open source server and client
- **ESRI products, http://esri.com**
  - Commercial GIS software
- **WorldWind, http://WorldWind.arc.nasa.gov**
  - Freeware, public, generates MANY million hits per day
- **SINTEF Globe, http://globe.sintef.no**
  - Public software, Java 3D
- **TopoFusion, http://www.topofusion.com**
  - GPS visualization 2D and 3D clients

Conclusion

- It is essential to have a data centric approach during the development of a high performance application that needs access to a massive dataset.
- A simple approach usually provides a better basis for higher performance than a complicated one.
- Access to the data available on the network is more convenient and sometimes faster than local data.
- Applications that use massive datasets on a distributed environments are possible, even for small projects.